AFRL-OSR-VA-TR-2013-0159

Internet attack traceback: cross-validation and pebble-trace

David Lee and Ten H. Lai
Ohio State University

April 2013
Final Report

DISTRIBUTION A: Approved for public release.

AIR FORCE RESEARCH LABORATORY
AF OFFICE OF SCIENTIFIC RESEARCH (AFOSR)
ARLINGTON, VIRGINIA 22203
AIR FORCE MATERIEL COMMAND

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE *(DD-MM-YYYY)*
28/02/2013

2. REPORT TYPE
FINAL REPORT

3. DATES COVERED *(From - To)*
01/04/2009 - 30/11/2012

4. TITLE AND SUBTITLE
Internet attack traceback: cross-validation and pebble-trace

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-09-1-0280

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
David Lee and Ten H. Lai

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
The Ohio State University
Department of Computer Science and Engineering,
2015 Neil Ave., Columbus, OH 43210

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AF Office of Scientific Research
875 N. Randolph St. Room 3112
Arlington, VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-OSR-VA-TR-2013-0159

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRIBUTION A: APPROVED FOR PUBLIC RELEASE

13. SUPPLEMENTARY NOTES

14. ABSTRACT

On the Internet, attackers often launch attacks through stepping-stones to steal confidential information from victims. Hiding behind stepping-stones, attackers thus avoid being traced back. In this project, the problem of Internet attack traceback was studied. A Pebbletrace scheme was proposed, which imbeds zero-day based Pebbleware in the stolen information and thereby enables one to trace back to the attacker's machine which has the stolen information.
A Pebbletrace prototype was built and focused on two cases: (1) the attacker steals a PDF file and (2) the attacker steals sensitive information through Zeus botnets. In the two cases, the project showed how to create Pebbleware automatically based on zero-day vulnerabilities, and how Pebbletrace reveals attackers whose machines are vulnerable to these zero-days.

15. SUBJECT TERMS
Traceback, zero-day vulnerability, Pebbleware, Pebbletrace, botnet.

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Ten H. Lai
U	U	U	SAR		19b. TELEPHONE NUMBER *(Include area code)* 614-292-2146

Reset

T · H · E
OHIO
STATE
UNIVERSITY

Internet attack traceback: cross-validation and pebble-trace

AFOSR Project Final Report

David Lee and Ten H. Lai

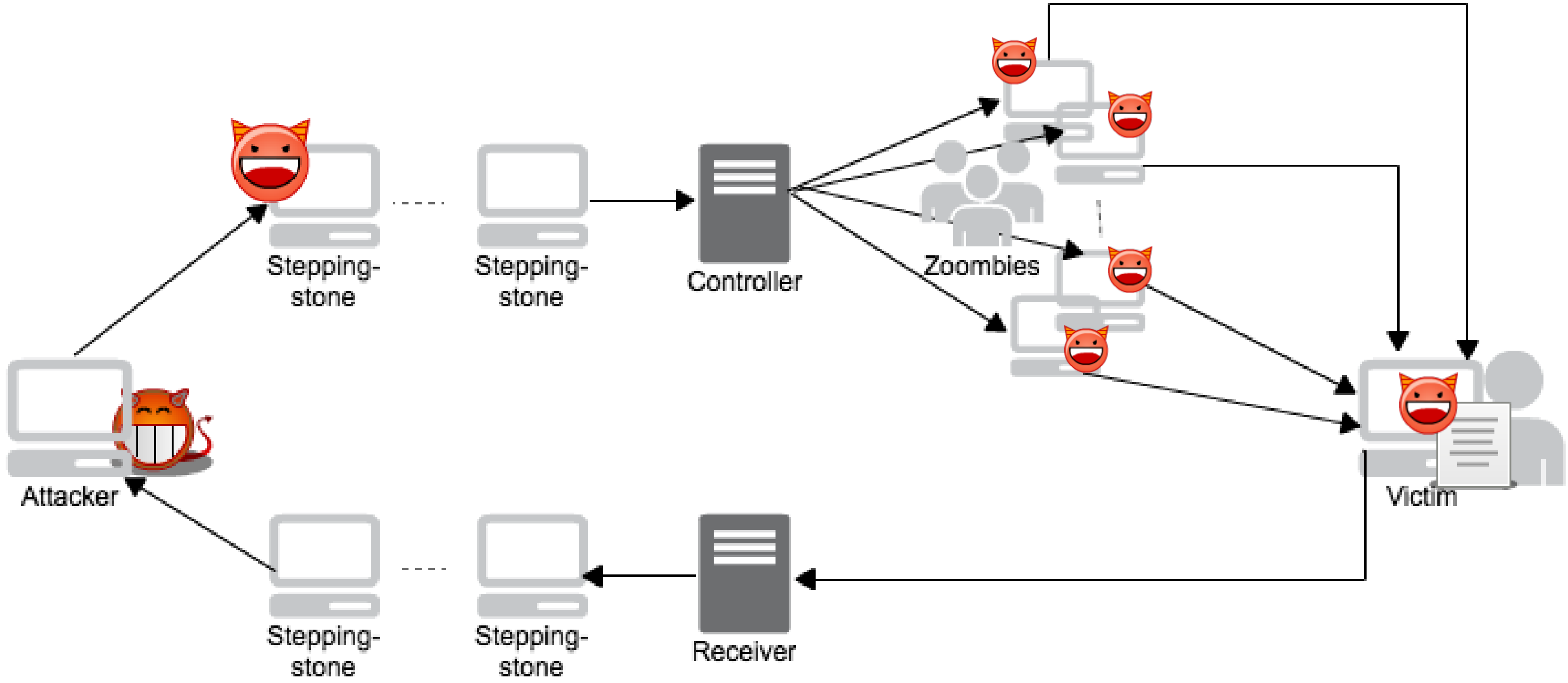
The Ohio State University

February 28, 2013

Outline

- Traceback Internet attacks
  - Problem
  - Challenges
- Our solution: Pebbletrace
  - Steal files
  - Steal information
- Conclusion

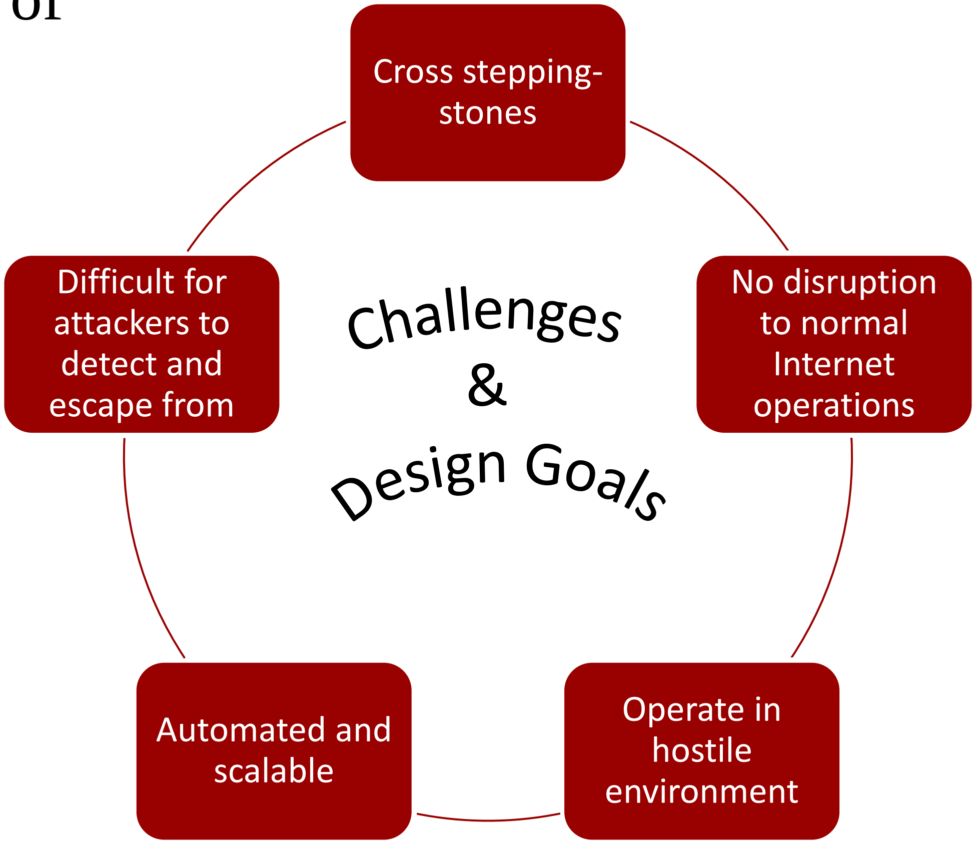
Model of Internet Attacks



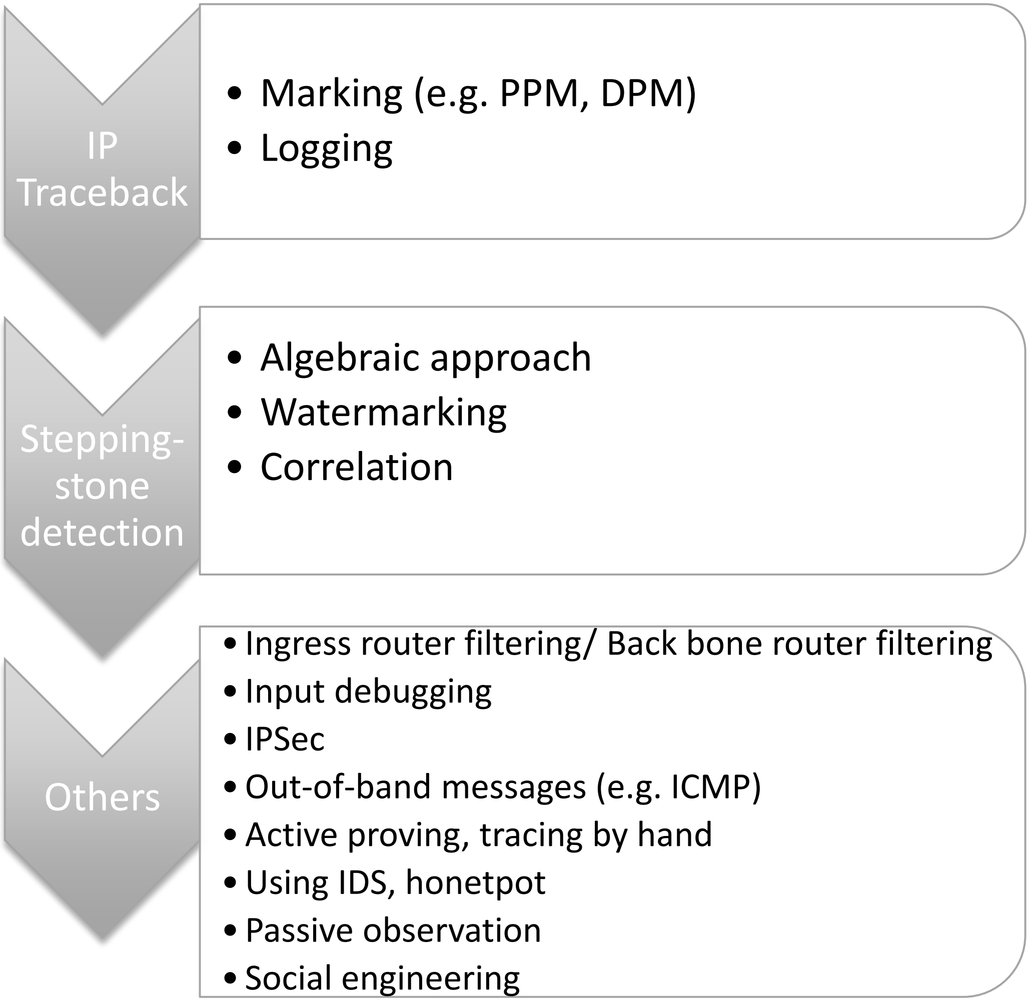
Attacker attempts to steal confidential files/information

Attack Traceback

Trace back to original source of attack across stepping stones


Cross stepping-stones
No disruption to normal Internet operations
Operate in hostile environment
Automated and scalable
Difficult for attackers to detect and escape from
Challenges
&
Design Goals

Prevalent Approaches



IP Traceback

- Marking (e.g. PPM, DPM)
- Logging

Stepping-stone detection

- Algebraic approach
- Watermarking
- Correlation

Others

- Ingress router filtering/ Back bone router filtering
- Input debugging
- IPSec
- Out-of-band messages (e.g. ICMP)
- Active proving, tracing by hand
- Using IDS, honetpot
- Passive observation
- Social engineering

None of them work for our traceback problem

Grimm's Fairy Tales "Hansel and Gretel"

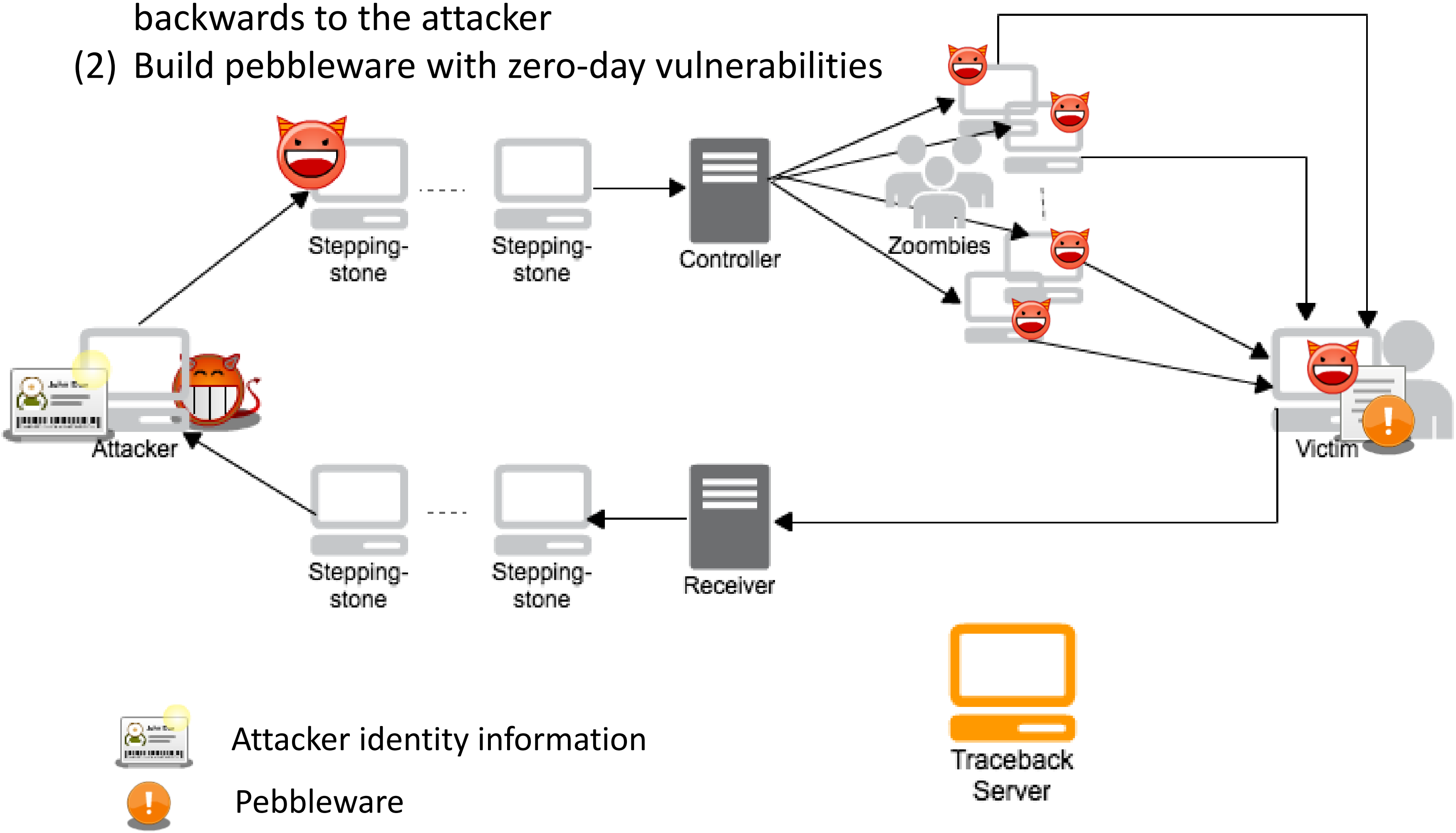
Courtesy by Childrensillustrators

Our Approach: Pebbletrace

Key idea:

(1) Take advantages of attacking traffic and trace backwards to the attacker

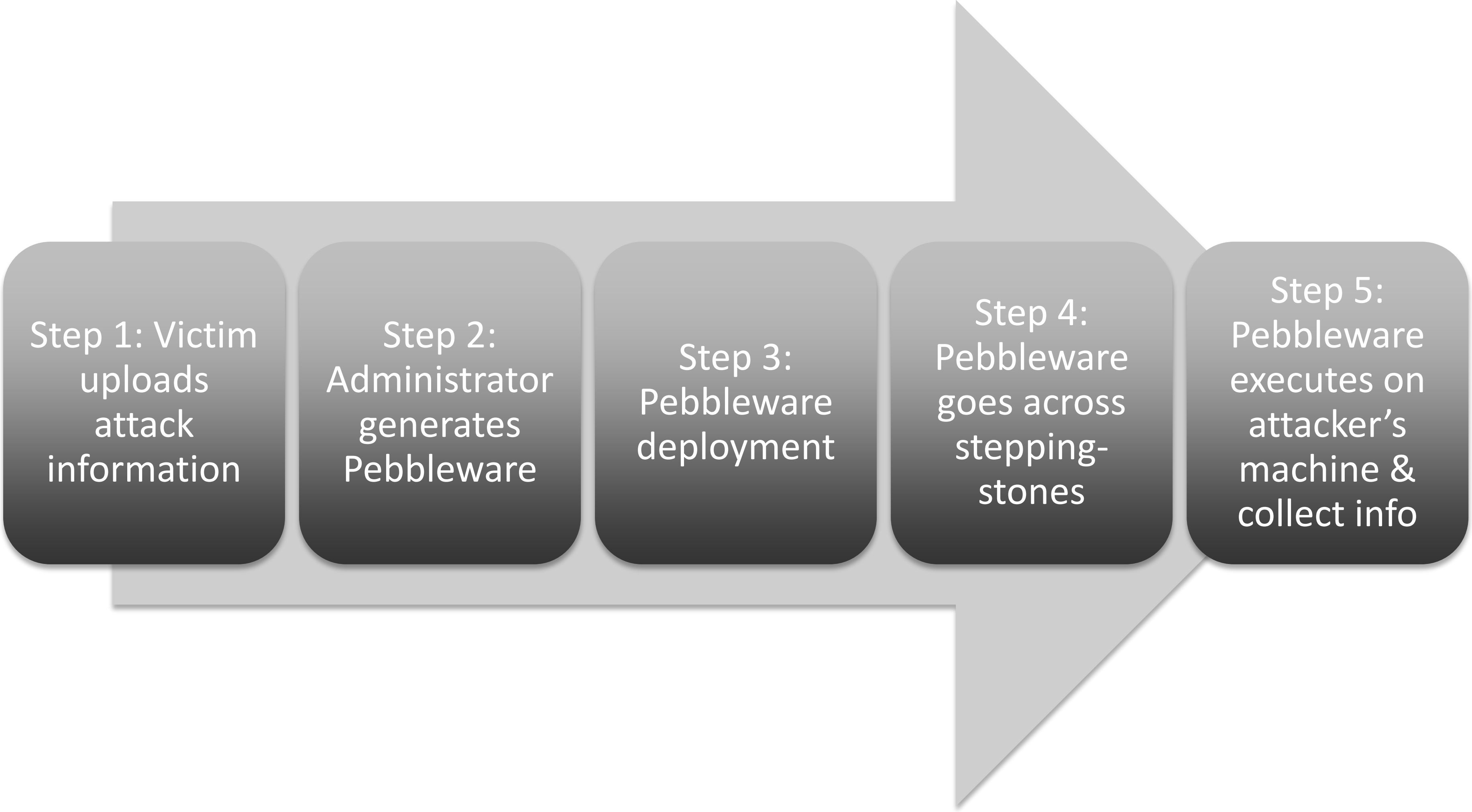
(2) Build pebbleware with zero-day vulnerabilities



Attacker identity information

Pebbleware

Steps of Pebbletrace



Step 1: Victim uploads attack information

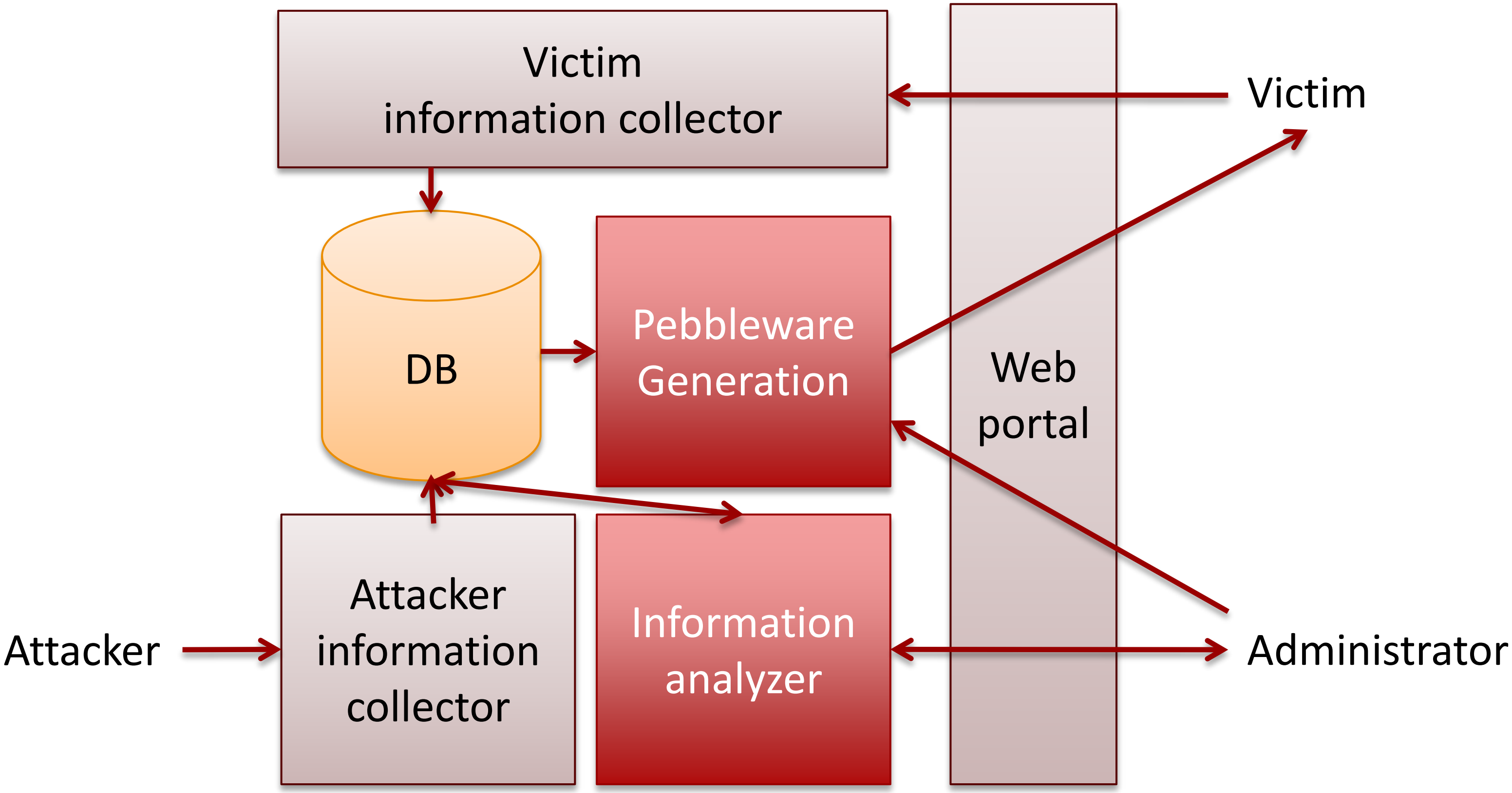
Step 2: Administrator generates Pebbleware

Step 3: Pebbleware deployment

Step 4: Pebbleware goes across stepping-stones

Step 5: Pebbleware executes on attacker’s machine & collect info

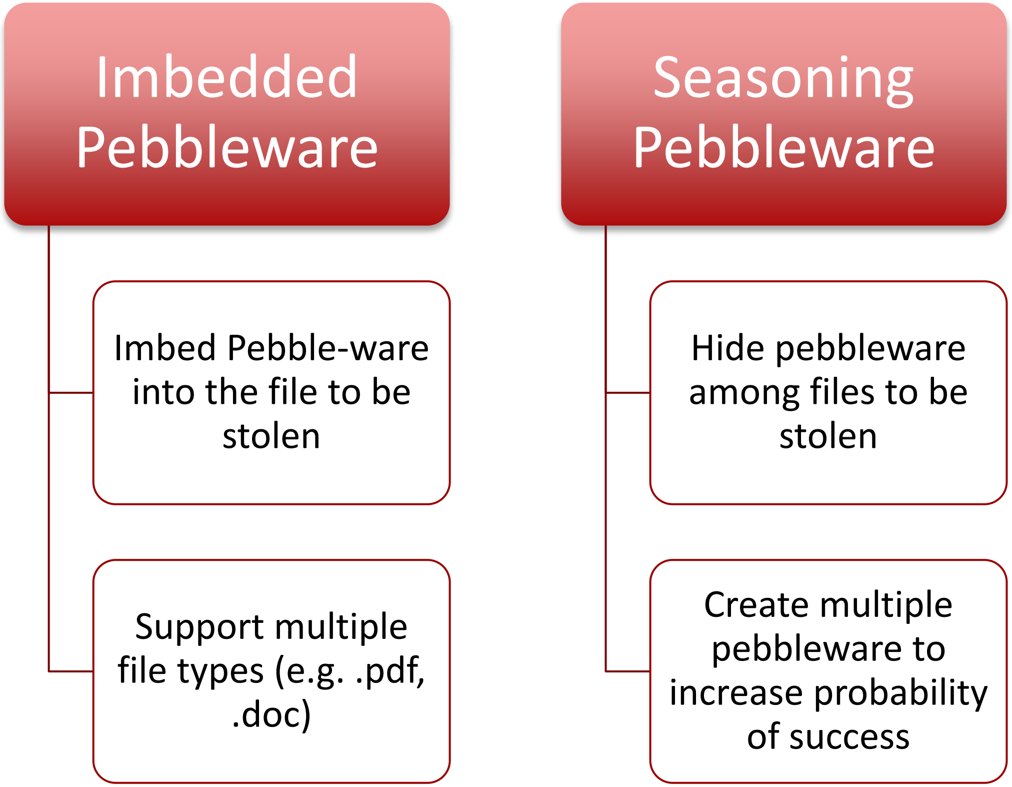
Architecture of Traceback Server



Problem 1: Attacker Steals Files

Key ideas

- Design Pebbleware based on client-side zero-day vulnerabilities
- Traceback attacker once the file containing Pebbleware is opened



Imbedded Pebbleware

Imbed Pebble-ware into the file to be stolen

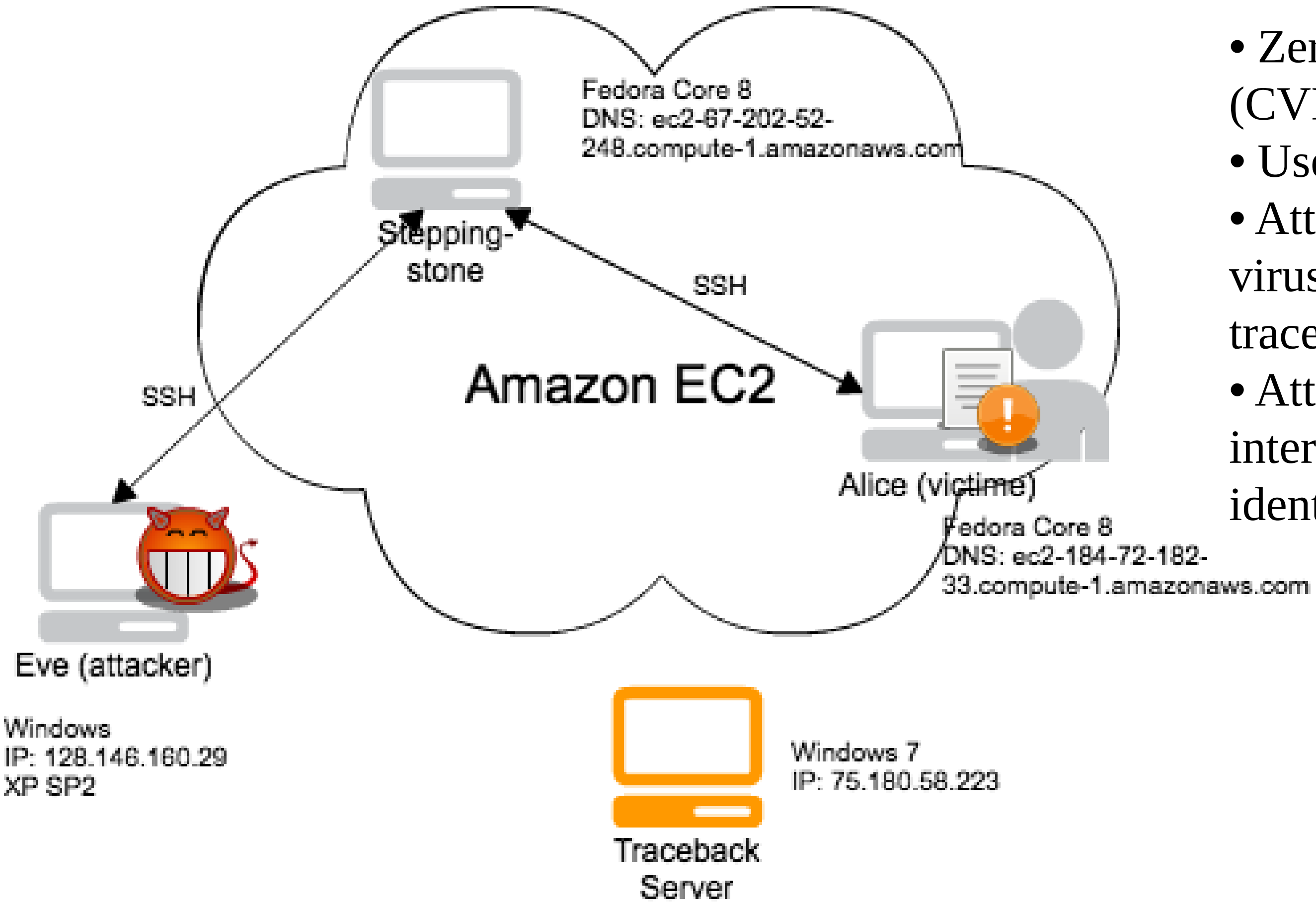
Support multiple file types (e.g. .pdf, .doc)

Seasoning Pebbleware

Hide pebbleware among files to be stolen

Create multiple pebbleware to increase probability of success

Case Study: Traceback File Theft at Amazon EC2



- Zero day: Adobe util.printf() (CVE-2008-2992)
- Use heap spray techinique
- Attacker’s firewall and anti-virus tools do not react to the traceback.
- Attacker’s IP, network interfaces, snapshot, etc. are identified.

Due to legal issues, the attack in the case study is constructed for study based on possible behaviors of real attackers, not accessible by public.

Problem 2: Attacker Steals Information

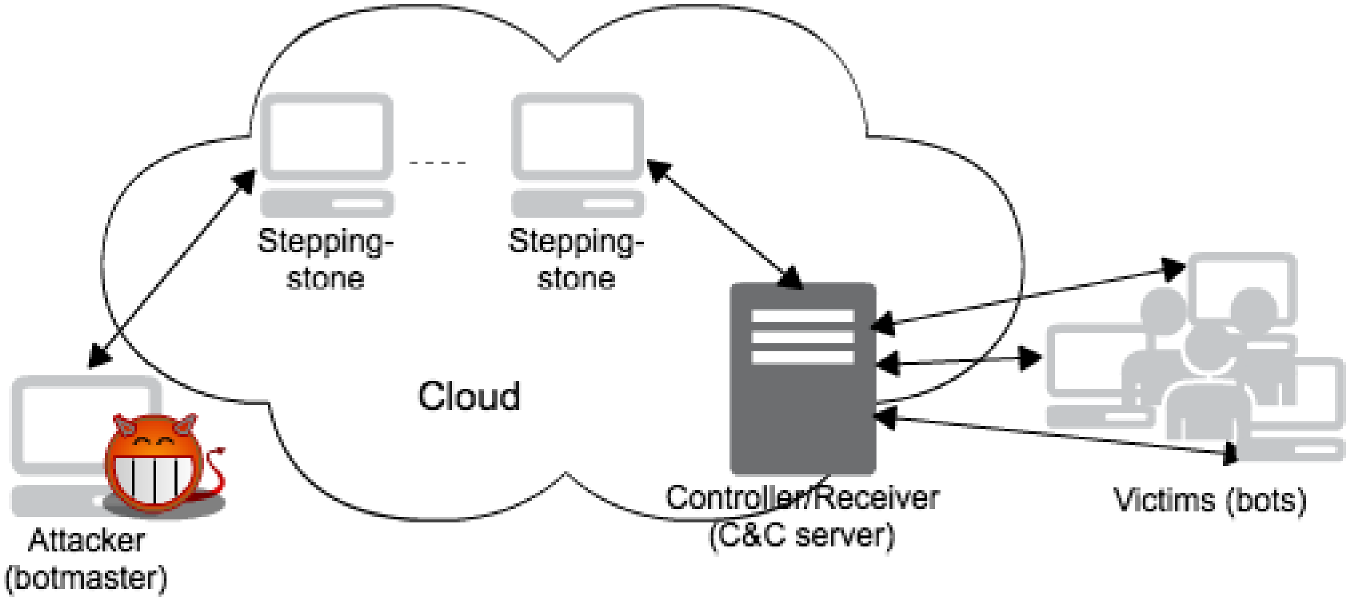
- Attacker steals confidential information (e.g. bank password) directly with hacker tools.
- How to imbed Pebbleware?

Focus on a scenario:

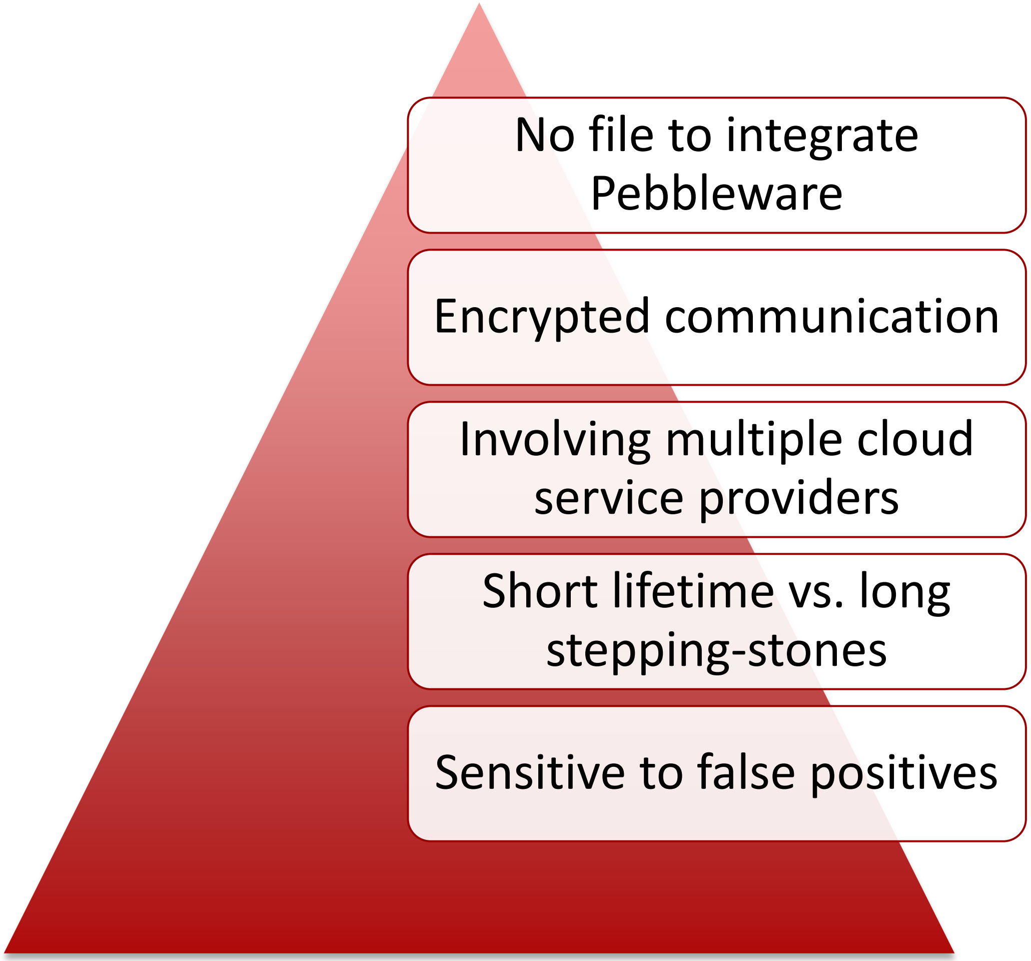
Traceback botmasters in cloud

Botnet Attacks in Cloud

- Scenarios
  - Communicating with victims
  - C&C servers and stepping stones in clouds
  - A centralized C&C server
  - Stepping stones: VPN, proxies and SSH tunneling
  - Symmetric encryption
    - RC4: Zeus, Feederbot; AES: Wraith, Waledac; DES: Ozdok
- Traceback: identify the botmaster behind stepping stones



Extra Challenges



No file to integrate Pebbleware

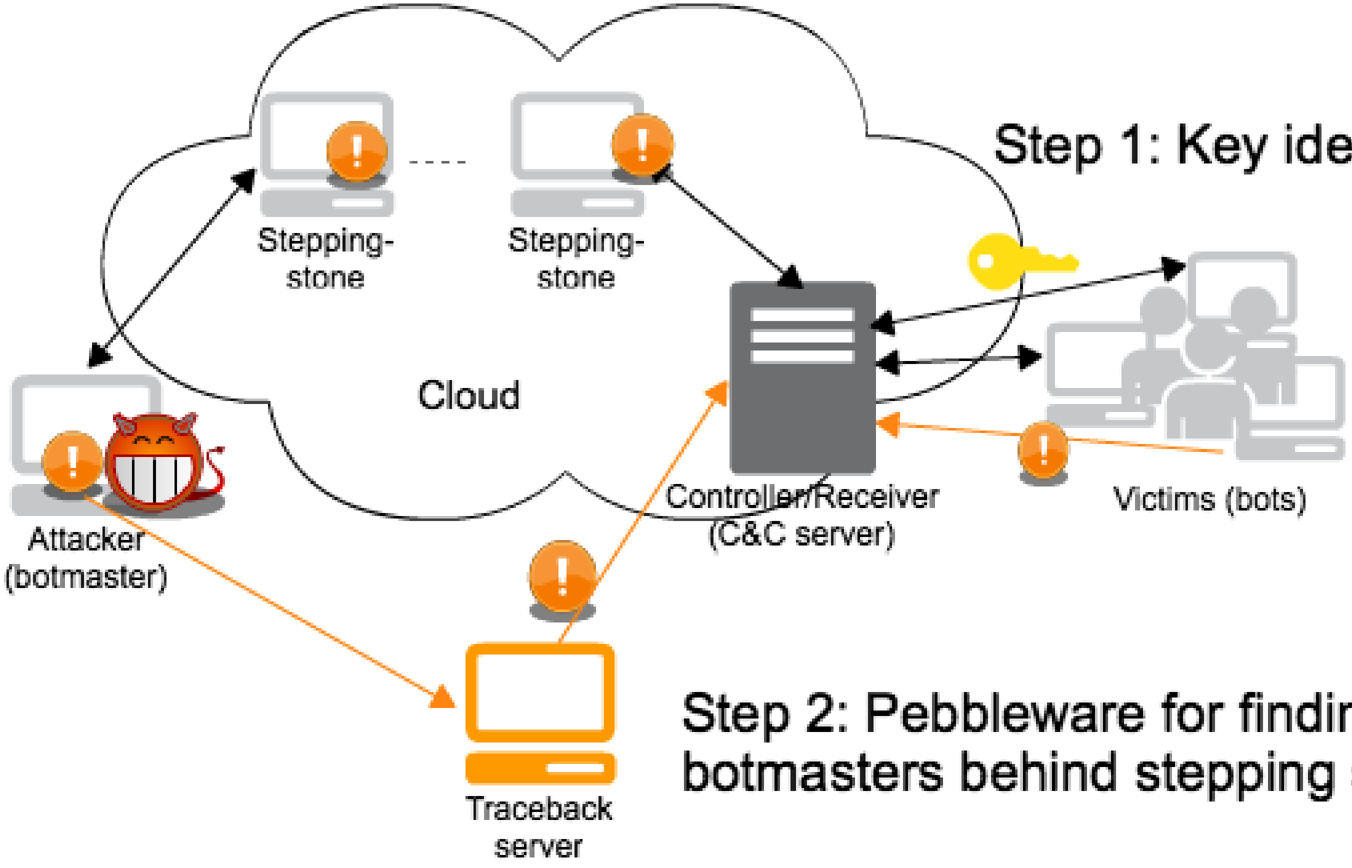
Encrypted communication

Involving multiple cloud service providers

Short lifetime vs. long stepping-stones

Sensitive to false positives

A General Approach to Pebbletracing Botmaster



Step 1: Key identification

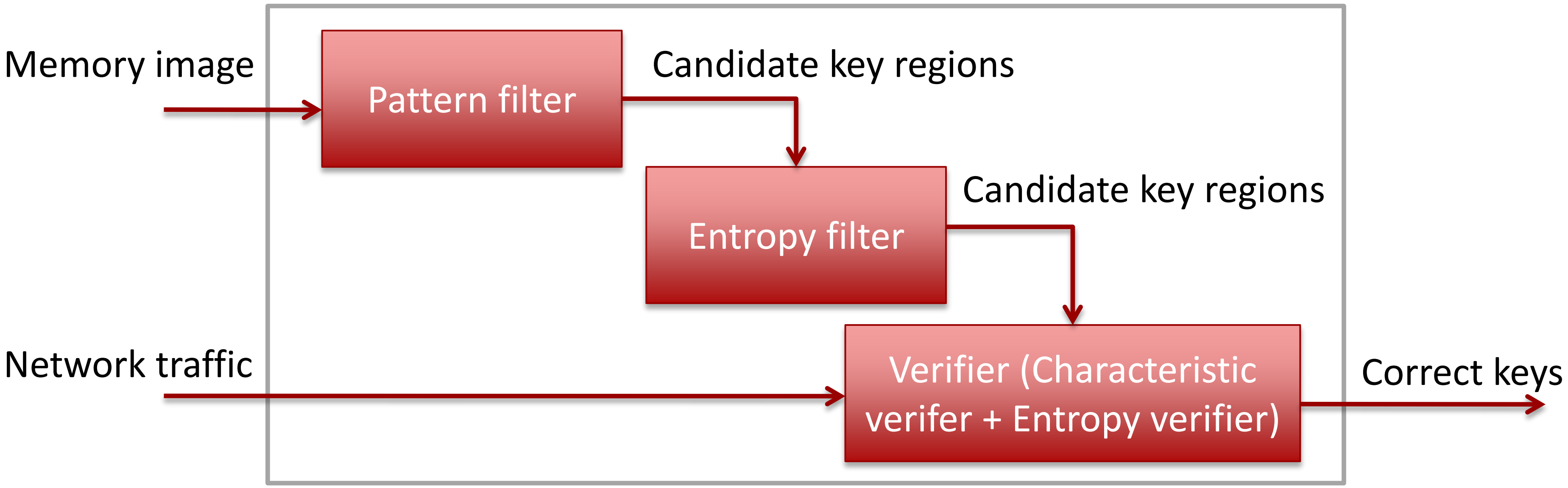
Step 2: Pebbleware for finding botmasters behind stepping stones

Step 1: Key Identification

- Finding the key given a memory image and encrypted traffic
- Constraints
  - No source code
  - Abnormal format patterns
  - Hard to verify candidate keys
  - Requiring low false positives

A Key Identification Scheme

- Observations
  - Fuzzy delimiter patterns may be available
  - Characteristics of symmetric keys
  - Randomness of ciphertext mostly from symmetric encryption schemes



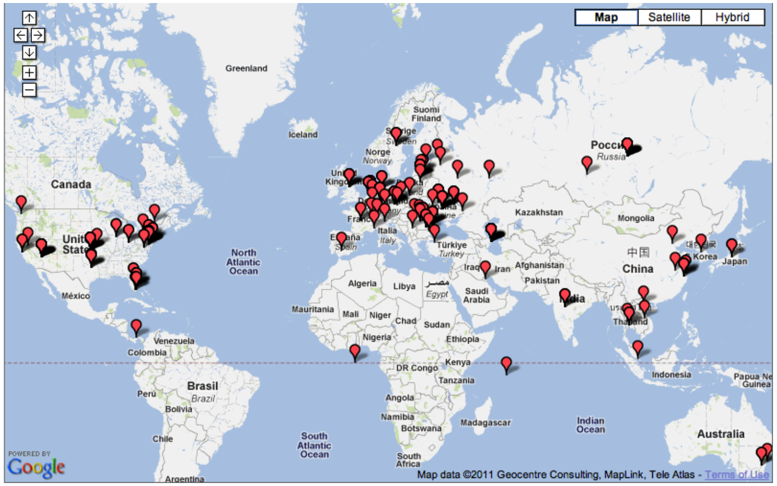
Step 2: Pebbleware for Finding Botmasters

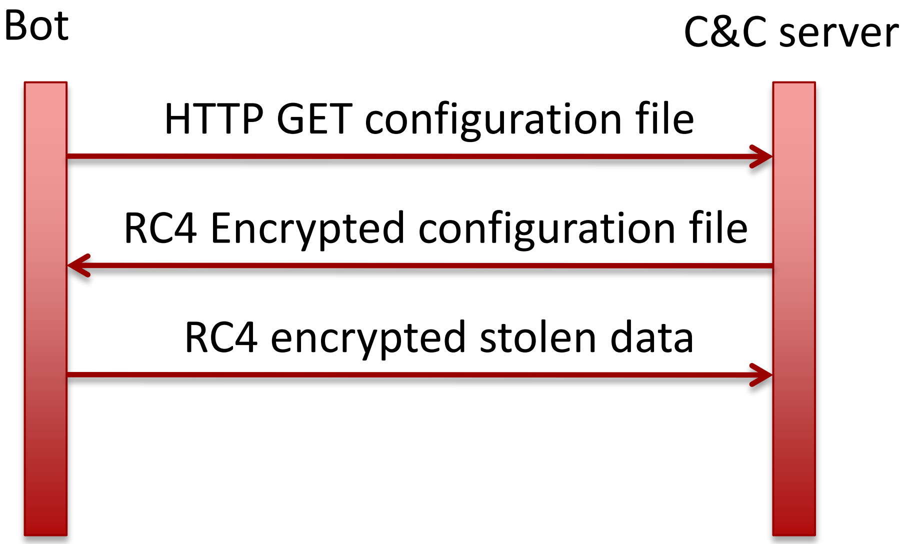
- Exploit zero-day vulnerabilities
  - Vulnerabilities of C&C servers
  - Client-side vulnerabilities
- Hard to select zero-days
- Hide Pebbleware into stealth traffic
  - Option 1: from victim
  - Option 2: from traceback server (e.g. pretend to be a victim)

Case Study: Traceback Zeus Botmaster in Opsource Cloud

Zeus: The king of bot

Distribution of Zeus C&C servers
(Jun 16, 2012 Courtesy by Zeustracker)





Bot

C&C server

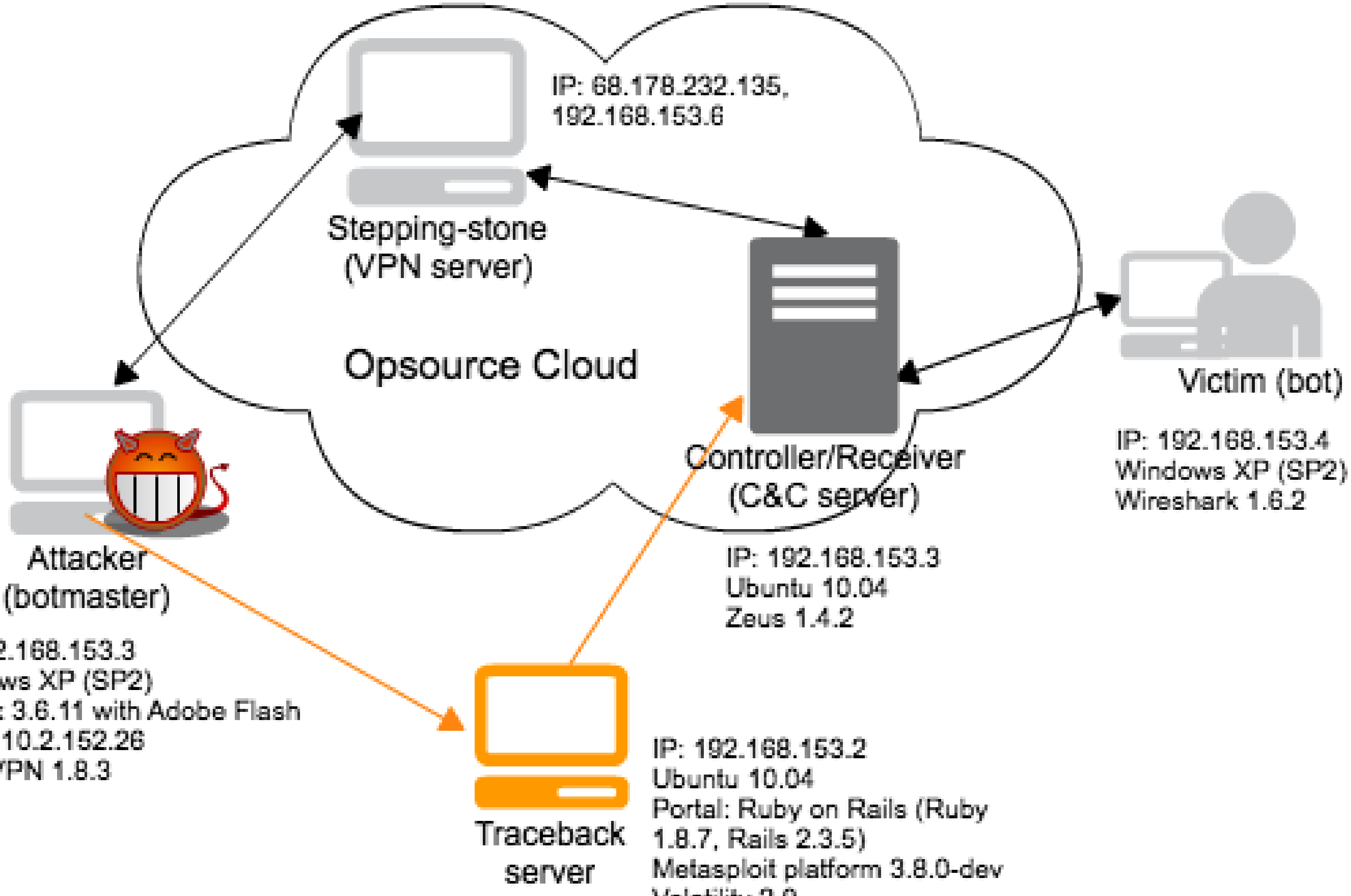
HTTP GET configuration file

RC4 Encrypted configuration file

RC4 encrypted stolen data

Basic Zeus protocol between bots and C&C server

Case Study: Traceback Zeus Botmaster in Opsource Cloud



IP: 68.178.232.135,
192.168.153.6

IP: 192.168.153.4
Windows XP (SP2)
Wireshark 1.6.2

IP: 192.168.153.3
Ubuntu 10.04
Zeus 1.4.2

IP: 192.168.153.3
Windows XP (SP2)
Firefox 3.6.11 with Adobe Flash
plugin 10.2.152.26
OpenVPN 1.8.3

IP: 192.168.153.2
Ubuntu 10.04
Portal: Ruby on Rails (Ruby
1.8.7, Rails 2.3.5)
Metasploit platform 3.8.0-dev
Volatility 2.0

Five Steps to Traceback Zeus Botmaster

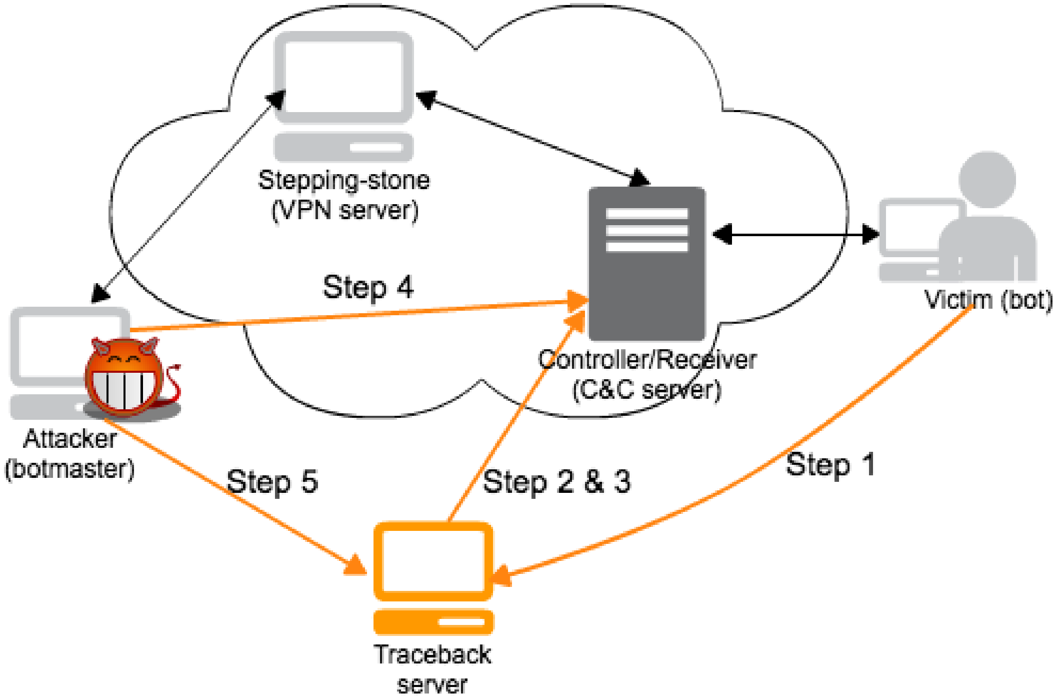
Step 1: Obtaining Information;

Step 2: Pebble 1---uploading the backdoored control panel;

Step 3: Pebble 1'---Replacing the control panel;

Step 4: Botmaster logins to C&C and is logged and redirected;

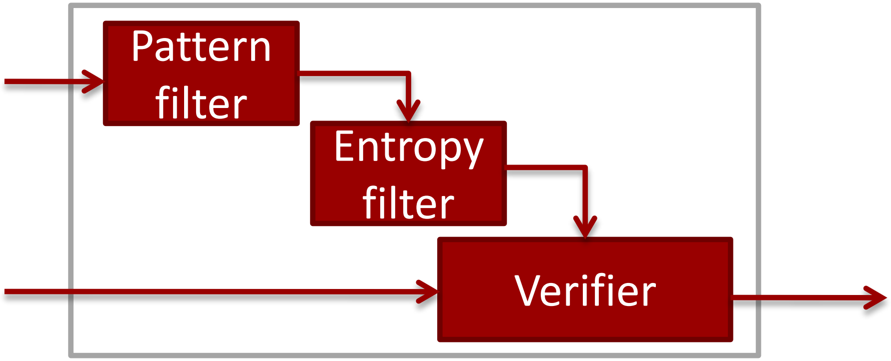
Step 5: Pebble 2---Penetrate stepping-stones collect attacker information.



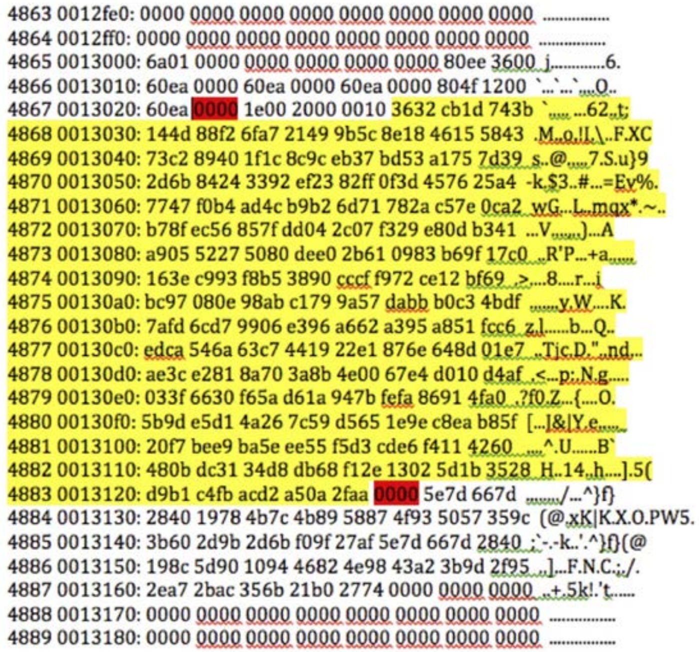
Step 1: Key Identification of Zeus Botnet

Identify RC4 keys of Zeus Bots

- Pattern filter: 2 zero bytes + 256--400 bytes + 2 zero bytes
- Entropy Analyzer: >7.5
- Verifier
  - Characteristics of key: a permutation of values in 0—255
  - Entropy verifier: the candidate key with largest entropy drop is the real key

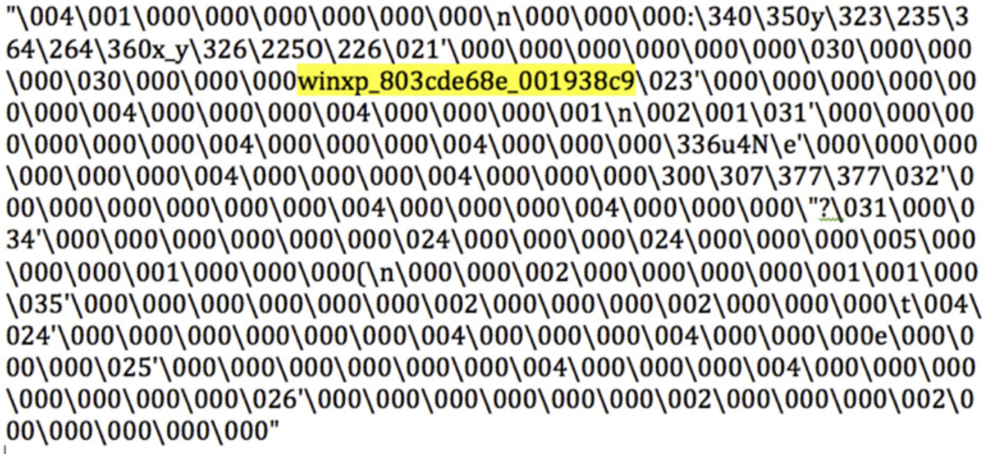


Identified key & Decrypted Traffic



```
4863 0012fe0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
4864 0012ff0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
4865 0013000: 6a01 0000 0000 0000 0000 0000 80ee 3600  j.............6.
4866 0013010: 60ea 0000 60ea 0000 60ea 0000 804f 1200  `...`...`....O..
4867 0013020: 60ea 0000 1e00 2000 0010 3632 cb1d 743b  `..... ...62..t;
4868 0013030: 144d 88f2 6fa7 2149 9b5c 8e18 4615 5843  .M..o.!I.\..F.XC
4869 0013040: 73c2 8940 1f1c 8c9c eb37 bd53 a175 7d39  s..@.....7.S.u}9
4870 0013050: 2d6b 8424 3392 ef23 82ff 0f3d 4576 25a4  -k.$3..#...=Ev%.
4871 0013060: 7747 f0b4 ad4c b9b2 6d71 782a c57e 0ca2  wG...L..mqx*.~..
4872 0013070: b78f ec56 857f dd04 2c07 f329 e80d b341  ...V.......)...A
4873 0013080: a905 5227 5080 dee0 2b61 0983 b69f 17c0  ..R'P...+a......
4874 0013090: 163e c993 f8b5 3890 cccf f972 ce12 bf69  .>....8....r...i
4875 00130a0: bc97 080e 98ab c179 9a57 dabb b0c3 4bdf  .......y.W....K.
4876 00130b0: 7afd 6cd7 9906 e396 a662 a395 a851 fcc6  z.l......b...Q..
4877 00130c0: edca 546a 63c7 4419 22e1 876e 648d 01e7  ..Tjc.D."..nd...
4878 00130d0: ae3c e281 8a70 3a8b 4e00 67e4 d010 d4af  .<...p:.N.g.....
4879 00130e0: 033f 6630 f65a d61a 947b fefa 8691 4fa0  .?f0.Z...{....O.
4880 00130f0: 5b9d e5d1 4a26 7c59 d565 1e9e c8ea b85f  [...J&|Y.e......_
4881 0013100: 20f7 bee9 ba5e ee55 f5d3 cde6 f411 4260   .....^.U......B`
4882 0013110: 480b dc31 34d8 db68 f12e 1302 5d1b 3528  H..14..h....].5(
4883 0013120: d9b1 c4fb acd2 a50a 2faa 0000 5e7d 667d  ......../...^}f}
4884 0013130: 2840 1978 4b7c 4b89 5887 4f93 5057 359c  (@.xK|K.X.O.PW5.
4885 0013140: 3b60 2d9b 2d6b f09f 27af 5e7d 667d 2840  ;`-.-k..'.^}f}(@
4886 0013150: 198c 5d90 1094 4682 4e98 43a2 3b9d 2f95  ..]...F.N.C.;./.
4887 0013160: 2ea7 2bac 356b 21b0 2774 0000 0000 0000  ..+.5k!.'t......
4888 0013170: 0000 0000 0000 0000 0000 0000 0000 0000  ................
4889 0013180: 0000 0000 0000 0000 0000 0000 0000 0000  ................
```

A detected S table of a Zeus bot

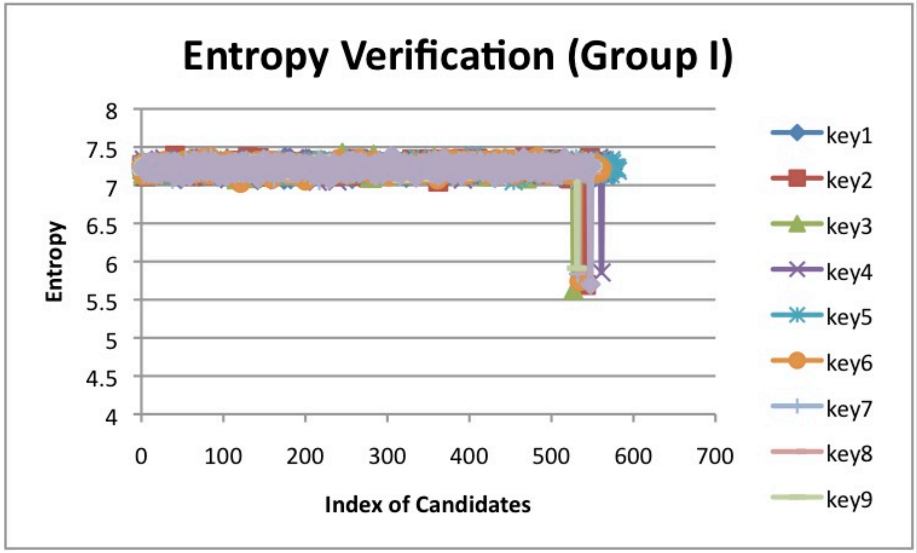


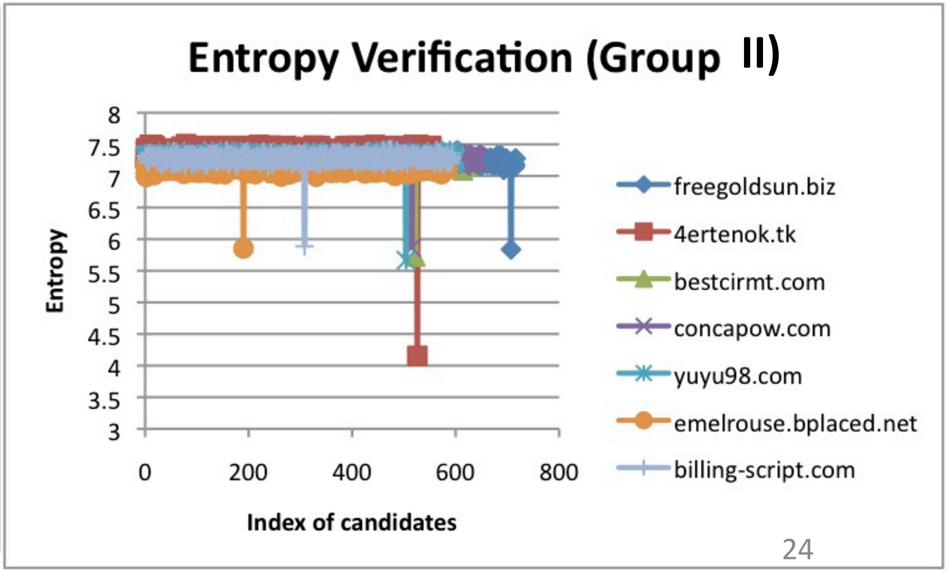
```
"\004\001\000\000\000\000\000\000\n\000\000\000:\340\350y\323\235\3
64\264\360x_y\326\225O\226\021'\000\000\000\000\000\000\030\000\000
\000\030\000\000\000winxp_803cde68e_001938c9\023'\000\000\000\000\00
0\000\004\000\000\000\004\000\000\000\001\n\002\001\031'\000\000\00
0\000\000\000\004\000\000\000\004\000\000\000\336u4N\e'\000\000\000
\000\000\000\004\000\000\000\004\000\000\000\300\307\377\377\032'\0
00\000\000\000\000\000\004\000\000\000\004\000\000\000\"?\031\000\0
34'\000\000\000\000\000\000\024\000\000\000\024\000\000\000\005\000
\000\000\001\000\000\000(\n\000\000\002\000\000\000\000\001\001\000
\035'\000\000\000\000\000\000\002\000\000\000\002\000\000\000\t\004\
024'\000\000\000\000\000\000\004\000\000\000\004\000\000\000e\000\0
00\000\025'\000\000\000\000\000\000\004\000\000\000\004\000\000\000
\000\000\000\000\026'\000\000\000\000\000\000\002\000\000\000\002\0
00\000\000\000\000"
```

A decrypted traffic of a Zeus bot

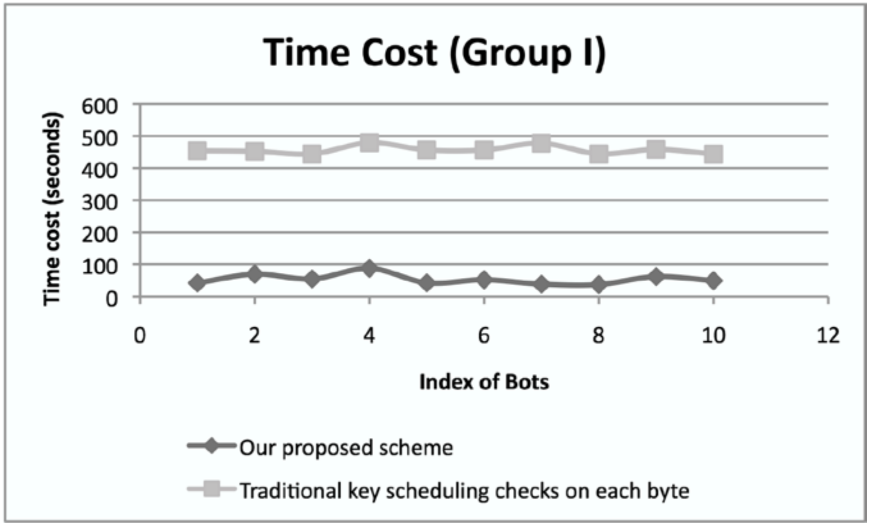
Performance of Entropy Verifier

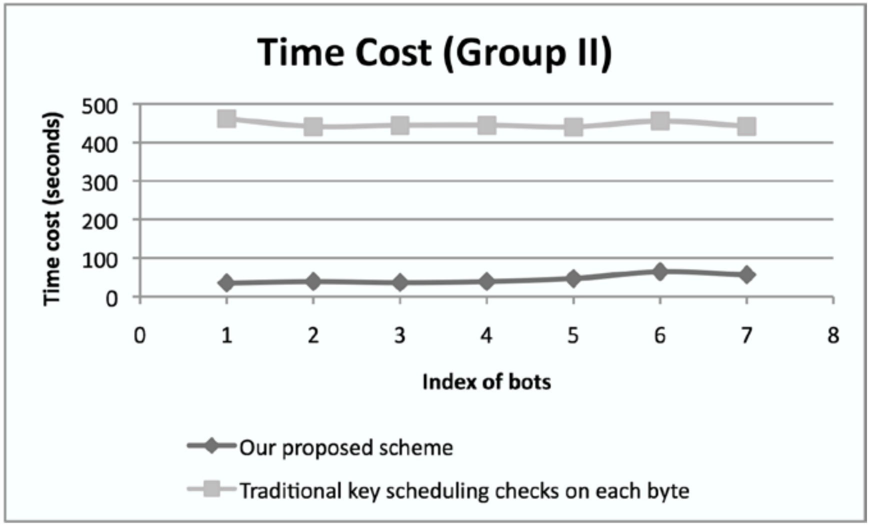
- Two groups of bots
  I. Homegrown
  II. Wild caught
- Outliers: the correct keys

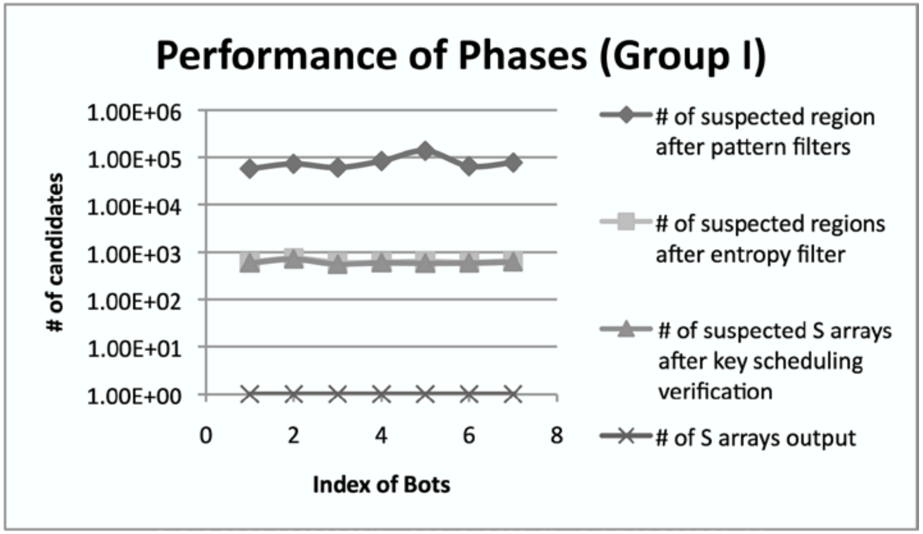


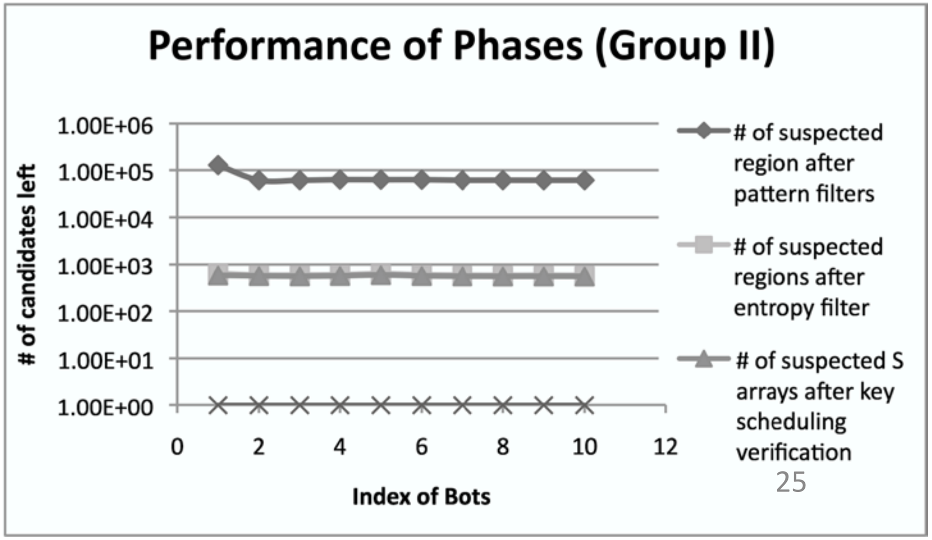


Performance of the Key Identification Scheme in Zeus Botnets

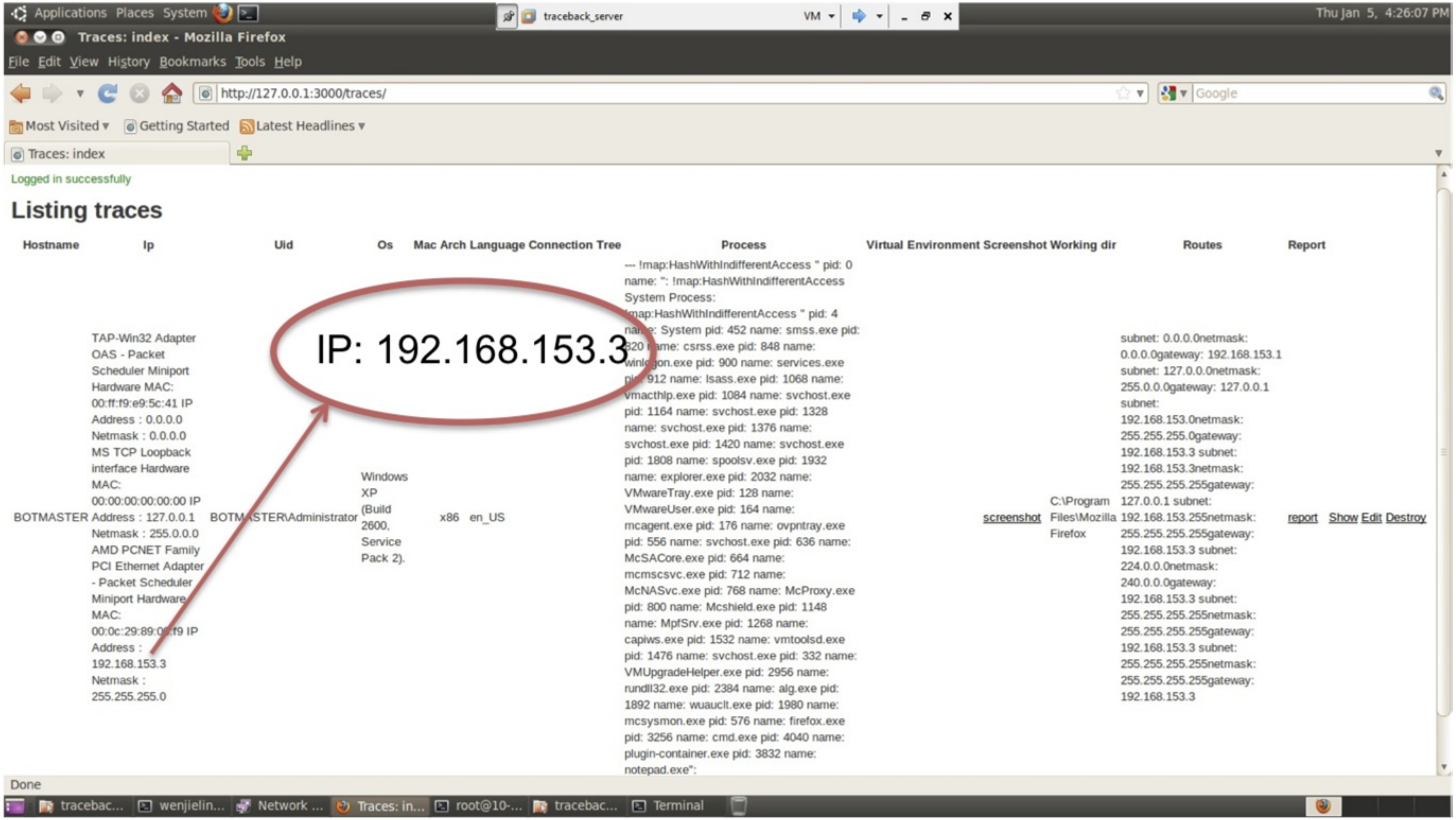

Time Cost (Group I)


Time Cost (Group II)


Performance of Phases (Group I)


Performance of Phases (Group II)

Attacker Information Detected



Logged in successfully

Listing traces

Hostname	Ip	Uid	Os	Mac	Arch	Language	Connection Tree	Process	Virtual Environment	Screenshot	Working dir	Routes	Report	
BOTMASTER	TAP-Win32 Adapter OAS - Packet Scheduler Miniport Hardware MAC: 00:ff:f9:e9:5c:41 IP Address : 0.0.0.0 Netmask : 0.0.0.0 MS TCP Loopback interface Hardware MAC: 00:00:00:00:00:00 IP Address : 127.0.0.1 Netmask : 255.0.0.0 AMD PCNET Family PCI Ethernet Adapter - Packet Scheduler Miniport Hardware MAC: 00:0c:29:89:0c:f9 IP Address : 192.168.153.3 Netmask : 255.255.255.0	BOTMASTER\Administrator	Windows XP (Build 2600, Service Pack 2).		x86	en_US		--- !map:HashWithIndifferentAccess " pid: 0 name: ": !map:HashWithIndifferentAccess System Process: !map:HashWithIndifferentAccess " pid: 4 name: System pid: 452 name: smss.exe pid: 820 name: csrss.exe pid: 848 name: winlogon.exe pid: 900 name: services.exe pid: 912 name: lsass.exe pid: 1068 name: vmacthlp.exe pid: 1084 name: svchost.exe pid: 1164 name: svchost.exe pid: 1328 name: svchost.exe pid: 1376 name: svchost.exe pid: 1420 name: svchost.exe pid: 1808 name: spoolsv.exe pid: 1932 name: explorer.exe pid: 2032 name: VMwareTray.exe pid: 128 name: VMwareUser.exe pid: 164 name: mcagent.exe pid: 176 name: ovpntray.exe pid: 556 name: svchost.exe pid: 636 name: McSACore.exe pid: 664 name: mcmscsvc.exe pid: 712 name: McNASvc.exe pid: 768 name: McProxy.exe pid: 800 name: Mcshield.exe pid: 1148 name: MpfSrv.exe pid: 1268 name: capiws.exe pid: 1532 name: vmtoolsd.exe pid: 1476 name: svchost.exe pid: 332 name: VMUpgradeHelper.exe pid: 2956 name: rundll32.exe pid: 2384 name: alg.exe pid: 1892 name: wuauclt.exe pid: 1980 name: mcsysmon.exe pid: 576 name: firefox.exe pid: 3256 name: cmd.exe pid: 4040 name: plugin-container.exe pid: 3832 name: notepad.exe":		screenshot	C:\Program Files\Mozilla Firefox	subnet: 0.0.0.0netmask: 0.0.0.0gateway: 192.168.153.1 subnet: 127.0.0.0netmask: 255.0.0.0gateway: 127.0.0.1 subnet: 192.168.153.0netmask: 255.255.255.0gateway: 192.168.153.3 subnet: 192.168.153.3netmask: 255.255.255.255gateway: 127.0.0.1 subnet: 192.168.153.255netmask: 255.255.255.255gateway: 192.168.153.3 subnet: 224.0.0.0netmask: 240.0.0.0gateway: 192.168.153.3 subnet: 255.255.255.255netmask: 255.255.255.255gateway: 192.168.153.3 subnet: 255.255.255.255netmask: 255.255.255.255gateway: 192.168.153.3	report	Show Edit Destroy

IP: 192.168.153.3

Conclusion

- Traceback Internet attacks
  - Attacker steals files
  - Attacker steals information
    - Traceback botmaster in clouds
- Future work
  - Attacker communicates with victims through social networks

Publications

- W. Lin and D. Lee, Traceback Attacks in Cloud— Pebbletrace Botnet, IEEE ICDCS SPCC, 2012
- Y. Hsu and D. Lee, Machine Learning for Implanted Malicious Code Detection with Incompletely Specified System Implementations, IEEE ICNP FIST, 2011
- G. Shu and D. Lee, "A Formal Methodology for Network Protocol Fingerprinting", IEEE Trans on Parallel and Distributed Systems, 2011
- Z. Liu, G. Shu and D. Lee, Instant Messaging Security, in Network Security, Administration and Management: Advancing Technologies and Practices, D. C. Kar and M. R. Syed, ed., IGI Global, 2010
- F. Yu, V. Gopalakrishnan, K. K. Ramakrishnan and D. Lee, “Loss-tolerant Real-time Content Integrity Validation for P2P Video Streaming”, COMSNETS 2009
- F. Yu and D. Lee, "Internet Attack Traceback - Cross-validation and Pebble Tracing", IEEE/DHS International Conference on Technologies for Homeland Security, May 2008
- Y. Hsu, G. Shu and D. Lee, “A Model-based Approach to Security Flaw Detection of Network Protocol Implementations”, IEEE ICNP Oct 2008
- G. Shu, Y. Hsu and D. Lee, “Fuzz Testing and Communications Protocol Security Flaws”, June FORTE 2008
- G. Shu, D. Chen, Z. Liu, N. Li, L. Sang and D. Lee, “VCSTC: Virtual Cyber Security Testing Capability – An Application Oriented Paradigm for Network Infrastructure Protection”, TESTCOM/FATES June 2008
- P. Pederson, D. Lee, G.-Q. Shu, D. Chen, Z. Liu, N. Li and L. Sang, "Virtual Cyber-Security Testing Capability for Large Scale Distributed Information Infrastructure Protection", IEEE/DHS International Conference on Technologies for Homeland Security, May 2008

Reference

[1] Sony Attacked Again, Passwords and Other Data Stolen http://threatpost.com/en_us/blogs/sony-attacked-again-passwords-and-other-data-stolen-060311.

[2] Paris G20 files stolen in cyber attack http://www.homelandsecuritynewswire.com/paris-g20-files-stolen-cyber-attack.

[3] Hacked: Data breach costly for Ohio State, victims of compromised info
http://www.thelantern.com/campus/hacked-data-breach-costly-for-ohio-state-victims-of-compromised-info-1.1831311.

[4] S. C. Lee and C. Shields, "Tracing the Source of Network Attack: A Technical, Legal and Societal Problem", In *Proc. of the 2001 IEEE Workshop on Information Assurance and Security*, June 2001.

[5] Guang Yao, Jun Bi, Zijian Zhou, "Passive IP traceback: capturing the origin of anonymous traffic through network telescopes", Proceedings of the ACM SIGCOMM 2010.

[6] Zeus (trojan horse), http://en.wikipedia.org/wiki/Zeus_(trojan_horse).

[7] Zeus: King of the Bots, http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/zeus_king_of_bots.pdf.

[8] Zeus Tracker, https://zeustracker.abuse.ch/.

[9] John the Ripper password cracker, http://www.openwall.com/john/.

[10] Taking over Zeus Botnet, http://xs-sniper.com/sniperscope/Zeus/CnC-Pwn-with-dir-traversal.txt

[11] Zeus botnets' Achilles' Heel makes infiltration easy, http://www.theregister.co.uk/2010/09/27/zeus_botnet_hijacking/.

[12] Metasploit, http://www.metasploit.com/.

[13] Metasploit PHP Executable Download and Execute payload, http://www.metasploit.com/modules/payload/php/download_exec.

[14] CVE-2011-0609, http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-0609

[15] RC4 http://en.wikipedia.org/wiki/RC4

[16] Binsalleeh, H., Ormerod, T., Boukhtouta, A., Sinha, P., Youssef, A., Debbabi, M., Wang, L., On the analysis of the Zeus botnet crimeware toolkit, Privacy Security and Trust (PST), 2010

[17] Adi Shamir and Nicko van Someren, Playing hide and seek with stored keys, Lecture Notes in Computer Science, 1999, Volume 1648/1999, 118-124